Supplementary material for:

Pattern classification of working memory networks reveals differential effects of methylphenidate, atomoxetine and placebo in healthy volunteers

Andre F. Marquand[1*], MSc, Sara De Simoni[1], MSc, Owen G. O'Daly[1], PhD, Steven C. R. Williams[1], PhD, Janaina Mourão-Miranda[2,1], PhD and Mitul A. Mehta[1], PhD

1: Department of Neuroimaging, Centre for Neuroimaging Sciences, Institute of Psychiatry, King's College London.

2: Centre for Computational Statistics and Machine Learning, Department of Computer Science, University College London.

*: Corresponding Author. Andre Marquand, Department of Neuroimaging, Centre for Neuroimaging Sciences, Box P089, Institute of Psychiatry. De Crespigny Park, London, SE58AF, United Kingdom. Tel: (+44) 203 228 3066. Fax: (+44) 203 228 2116. Email: andre.marquand@kcl.ac.uk

File Description: Supplementary Methods and Supplementary Results (13 pages total)

Supplementary Methods

Visual analogue scales

Individual items from the visual analogue scales (VAS) were collapsed to reflect subjective factors 'alertness' and 'tranquillity' (Herbert et al., 1976). Scales used to calculate 'alertness' were: alert–drowsy, strong–feeble, muzzy–clear headed, well coordinated–clumsy, lethargic–energetic, mentally slow–quick witted, attentive–dreamy, incompetent–proficient and interested–bored. VAS items used to calculate 'tranquillity' were: calm–excited, contented–discontented, troubled–tranquil, tense–relaxed, happy–sad, antagonistic–amicable and withdrawn–gregarious. The words on the right were scored as 100.

Gaussian process classification

We only provide a brief introduction to GPC inference here, but more detail can be found elsewhere (Rasmussen and Williams, 2006; Marquand et al., 2010b). Formally, a Gaussian process (GP) is the generalization of the multivariate Gaussian distribution to infinitely many dimensions (where any finite number are multivariate Gaussian) and can be uniquely described by its mean ($m(x)$) and covariance ($k(x,x')$) functions: $GP \sim N(m(x), k(x,x'))$. Given a set of 'training' data: $D = \{\mathbf{X}_{n\times d}, \mathbf{y}_{n\times 1}\} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where the $\mathbf{x}_i$ are *d*-dimensional input vectors and the y_i are binary class labels satisfying $y_i \in \{1, -1\}$, GP models may be used to learn statistical properties of the training set that enables accurate prediction of the label of unseen data points (the 'test set') using Bayesian probability theory. Predictions take the form of class probabilities: $p(y_i^* = C|\mathbf{x}_i^*, D)$, which describe the probability that data sample $\mathbf{x}_i^*$ belongs to class *C* given the training data.

There are two equivalent perspectives on GP inference, referred to as the 'weight-' and 'function space' views. From the weight space view, linear GPC can be considered a Bayesian extension of logistic regression where the probability of membership of class 1 is derived by squashing an unbounded linear function of the input vectors and a weight vector ($\mathbf{w}$) through a sigmoidal 'likelihood' function (σ), i.e.: $p(y_i = 1|\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}) = \sigma(f_i)$. This ensures predictions lie on the unit interval and so have a valid probabilistic interpretation. In this paper we use the probit likelihood: $\sigma(z) = \Phi(z) = \int_{-\infty}^{z} N(u|0,1)du$.[1] In ordinary logistic regression the weights are estimated by maximum likelihood, which can be prone to overfitting and is not appropriate in ill-posed problem domains where the input dimensionality greatly exceeds the number of samples (e.g. neuroimaging data). In GPC inference this is solved by first applying a zero-mean Gaussian prior to the weights, then computing the posterior weight distribution by the rules of probability (especially Bayes rule).

It turns out to be more convenient to adopt the function space view of GP modelling, where inference proceeds by applying a zero-mean Gaussian prior directly to the f_i, which are viewed as a latent function that models relationships using the data. From this perspective, the weights are integrated out as nuisance parameters and inference proceeds by computing the posterior function distribution by Bayes rule. GPC inference can thus be divided into two steps: (1) use Bayes' rule to compute the posterior function distribution and (2) compute the posterior expectation of the test point to produce a prediction. Collecting all function values into a vector ($\mathbf{f}$), Bayes' rule can be written as:

$$p(\mathbf{f}|D, \boldsymbol{\theta}) = \frac{p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{y}|\mathbf{f})}{p(D|\boldsymbol{\theta})} = \frac{N(\mathbf{f}|\mathbf{0}, \mathbf{K})}{p(D|\boldsymbol{\theta})} \prod_{i=1}^{m} \sigma(y_i f_i) \qquad \text{(S1)}$$

Here, $N(\mathbf{0}, \mathbf{K})$ describes the prior over the latent function and we have factorised the likelihood over training samples. We have also written each likelihood term as $p(y_i|f_i) = \sigma(y_i f_i)$ owing to the symmetry of the probit likelihood. Model hyperparameters are denoted by $\boldsymbol{\theta}$, and can be set by maximising the marginal likelihood (the denominator in equation S1). The posterior in equation S1 are both analytically intractable, but can be approximated by a Gaussian: $q(\mathrm{f}|D, \boldsymbol{\theta}) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

[1] This implies that technically, the implementation of GPC employed here should be considered an extension of 'probit regression' not logistic regression.

where the approximate parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are computed using the expectation propagation algorithm (Minka, 2001; Rasmussen and Williams, 2006). Once the approximate posterior has been computed, it can then be used to compute: (1) the marginal likelihood and (2) the approximate posterior for the test case. Following Kuss and Rasmussen (2005), the latter can be computed by:

$$\begin{aligned} q(f^*|D, \mathbf{x}^*, \boldsymbol{\theta}) &= N(f^*|\mu^*, \sigma^{2^*}) \\ \mu^* &= \mathbf{k}^{*\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{\mu} \\ \sigma^{2^*} &= k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{*\mathrm{T}}(\mathbf{K}^{-1} - \mathbf{K}^{-\mathbf{1}}\boldsymbol{\Sigma}\mathbf{K}^{-1})\mathbf{k}^* \end{aligned}$$

Finally, predictions are made by computing the posterior expectation of the latent function at the test point:

$$\begin{aligned} p(y_* = 1|D, \mathbf{x}^*) &= \int \Phi(f^*) q(f^*|D, \boldsymbol{\theta}, \mathbf{x}^*) df^* \\ &= \Phi\left(\frac{\mu^*}{\sqrt{1 + \sigma^{2^*}}}\right) \end{aligned} \qquad \text{(S2)}$$

We used a customised version of the Gaussian processes for machine learning toolbox (www.gaussianprocess.org/gpml) for all GPC inference.

GPC covariance function

A crucial component of GPC inference is the prior covariance function (or kernel), which measures similarity between all data points. Like SVM, GPC supports the specification of non-linear kernels, but in this paper, we use a linear kernel that can be specified as:

$$\mathbf{K} = \frac{1}{l^2}\mathbf{X}\mathbf{X}^{\mathrm{T}} + b \qquad \text{(S3)}$$

Where the model hyperparameters are: l^2, (a regularisation hyperparameter) and *b* (a bias term) (Marquand et al., 2010b). Using such a linear kernel helps to prevent overfitting and allows direct extraction of the weight vector as an image, which is important to estimate the contribution of each voxel to the decision function and is essential for recursive feature elimination, as described in the next section.

Recursive Feature Elimination

Recursive feature elimination(RFE; Guyon et al., 2002) is a backward elimination feature selection approach that aims to find a parsimonious set of discriminating features by iteratively removing the least informative features. Here we present an adaptation of RFE for GP classifiers ('GPC-RFE'), introduced in Marquand et al. (2010a). RFE involves repeatedly training a classifier and at each iteration ranking features and removing a subset of the lowest ranking features, which continues until no features remain. The predictive performance of the classifier is measured at each stage of feature removal (on an independent sample), allowing an optimal number of features that maximises predictive performance to be selected. For the ranking criterion, we use the square of the maximum aposteriori estimate of the weight vector (i.e. $\{\hat{w}_j^{\mathbf{2}}\}_{j=1}^{d}$), which using vector notation can be computed by:

$$\hat{\mathbf{w}} = \frac{1}{l^2}\mathbf{X}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} \tag{S4}$$

We remove a fixed number of features at each iteration (2% of cerebral voxels). This value was chosen empirically to provide fine-grained control over the number of features retained in reasonable computation time, but in practice we found similar results were obtained using a range of step sizes.

Determining the optimal number of features

We used nested leave-one-out cross-validation (LOO-CV) to determine the optimal number of features as described in the main text (Algorithm S1). For optimal performance, it is important to be able to accurately measure classifier performance at different stages of feature removal. RFE is usually applied to SVM classifiers, where classification accuracy is the most common measure of classifier performance. However, classification accuracy is a coarse measure that does not consider the confidence of a classifier's predictions. For probabilistic classifiers such as GPC, classification accuracy is therefore suboptimal. Following Guyon et al. (2002) who used several metrics of classifier quality at each stage of feature elimination, we use classification accuracy and an additional metric known as 'target information' (Rasmussen and Williams, 2006) that includes information about predictive confidence in addition to whether predictions are right or wrong (e.g. strongly penalising confident misclassifications) which helps: (1) to differentiate voxel subsets that would otherwise produce equal LOO-CV accuracy and (2) to reduce the variability of the size of the optimal feature set across LOO-CV folds. For binary classifiers, target information can be computed by equation 3 (see also: Rasmussen and Williams, 2006, ch. 3):

$$I = \frac{1}{n}\left[\sum_{i\,\in C1} \log_2(p(y_i^* = 1|\mathbf{x}_i^*)) + \sum_{j\,\in C2} \log_2(1 - p(y_j^* = -1|\mathbf{x}_j^*))\right] + H \qquad \text{(S5)}$$

C1 and *C2* denote class 1 and 2 respectively, *H* is the entropy of the test set and we have omitted explicit dependence on the training dataset for clarity. Note that *H* = 1 if the number of elements belonging to class 1 and 2 are equal (which is always true here).

Visualisation of the differential activity pattern

As described in the main text, to visualise the differential activity pattern for each classifier, we did not visualise the classifier weights, but instead employed a mapping approach that permits visualisation of the relative class distribution (Marquand et al., 2010b). Thus, multivariate brain maps can be computed by: $\mathbf{g} = 1/l^2\mathbf{X}^{\mathrm{T}}\boldsymbol{\mu}$, where $\boldsymbol{\mu}_{n\times1}$ is the mean of the latent function evaluated at each data point.

For clarity, a summary of GPC-RFE is presented in algorithm S1

Algorithm S1: GPC-RFE

```
Input:  N number of subjects
        V: total number of voxels
        S: step size
        $\mathbf{X}_1$: data matrix (class 1)
        $\mathbf{X}_2$: data matrix (class 2)
        $\mathbf{y}$: data labels

for i = 1,..,N        // Nested LOO-CV loop
  // Parameter optimization
  $\mathbf{v} = [1,..,V]^{\mathrm{T}}$   // voxels to include
  $\mathbf{X}_{test} = [\mathbf{X}_1(i,:)^{\mathrm{T}}\ \mathbf{X}_2(i,:)^{\mathrm{T}}]^{\mathrm{T}}$
  for j = 1,…, i-1, i+1,…, N
    k = length($\mathbf{v}$)
    while k > 1
      $\mathbf{X}_{VALID} = [\mathbf{X}_1(j,\mathbf{v})^{\mathrm{T}}\ \mathbf{X}_2(j,\mathbf{v})^{\mathrm{T}}]^{\mathrm{T}}$
```

```
            $\mathbf{X}_{TRAIN} = [\mathbf{X}_1(\backslash ij,\mathbf{v})^T\ \mathbf{X}_2(\backslash ij,\mathbf{v})^T]^T$
            train/test GPC using $\mathbf{X}_{TRAIN}/\mathbf{X}_{VALID}$
            save validation predictions
            compute $\hat{\mathbf{w}}$ (Equation S3)
            remove S voxels having lowest $\hat{w}_l^2$ from $\mathbf{v}$
        end while
    end for
    // now compute target information and accuracy on the validation set ($\mathbf{i}_{VALID}$ and $\mathbf{a}_{VALID}$) using
    // saved predictions and determine the optimal number of voxels ($\mathbf{v}_{OPT}$ ) for this LOO-CV fold
    Compute $\mathbf{a}_{VALID}$ and $\mathbf{i}_{VALID}$ for all k (Equation S5)
    $\mathbf{v}_{OPT}(i) = \text{mean}(\text{argmax}\ (\mathbf{a}_{VALID}), \text{argmax}(\mathbf{i}_{VALID}\ ))$

    // Testing
    $\mathbf{v} = [1,..,V]^T$
    $\mathbf{X}_{TRAIN} = [\mathbf{X}_1(\backslash i,:)^T\ \mathbf{X}_2(\backslash i,:)^T]^T$
    train GPC using $\mathbf{X}_{TRAIN}$
    compute $\hat{\mathbf{w}}$ (Equation S4)
    remove $\mathbf{v}_{OPT}(i)$ voxels with lowest $\hat{w}_l^2$ from $\mathbf{v}$
    $\mathbf{X}_{TRAIN} = [\mathbf{X}_1(\backslash i,\mathbf{v})^T\ \mathbf{X}_2(\backslash i,\mathbf{v})^T]^T$
    $\mathbf{X}_{TEST} = [\mathbf{X}_1(i,\mathbf{v})^T\ \mathbf{X}_2(i,\mathbf{v})^T]^T$
    train/test GPC using $\mathbf{X}_{TRAIN}/\mathbf{X}_{TEST}$
    save test predictions
end for
// finally compute performance metrics
compute $\mathbf{a}_{TEST}$ and $\mathbf{i}_{TEST}$ (Equation S5)
```

Supplementary Results and Discussion

Performance measures

A summary of mean RT and accuracy for each drug and reward condition of the WM task during each is provided in Table S 1.

	Placebo		**Atomoxetine**		**Methylphenidate**	
	Rewarded	**Non-rewarded**	**Rewarded**	**Non-rewarded**	**Rewarded**	**Non-rewarded**
Mean (SEM) Reaction Time (ms)	1022 (38)	1030 (41)	1020 (36)	1031 (28)	1032 (27)	1016 (18)
Mean (SEM) Accuracy (%)	83.00 (0.02)	76.00 (0.03)	80.30 (0.02)	80.00 (0.02)	83.33 (0.02)	74.65 (0.04)

Table S 1: Reaction time and accuracy for the rewarded WM task. Mean (SEM) of 15 subjects.

Distribution maps: task versus baseline (control)

Whole-brain distribution maps derived from the classifiers trained to separate each WM component from baseline using non-rewarded trials are presented in Figure S 1.

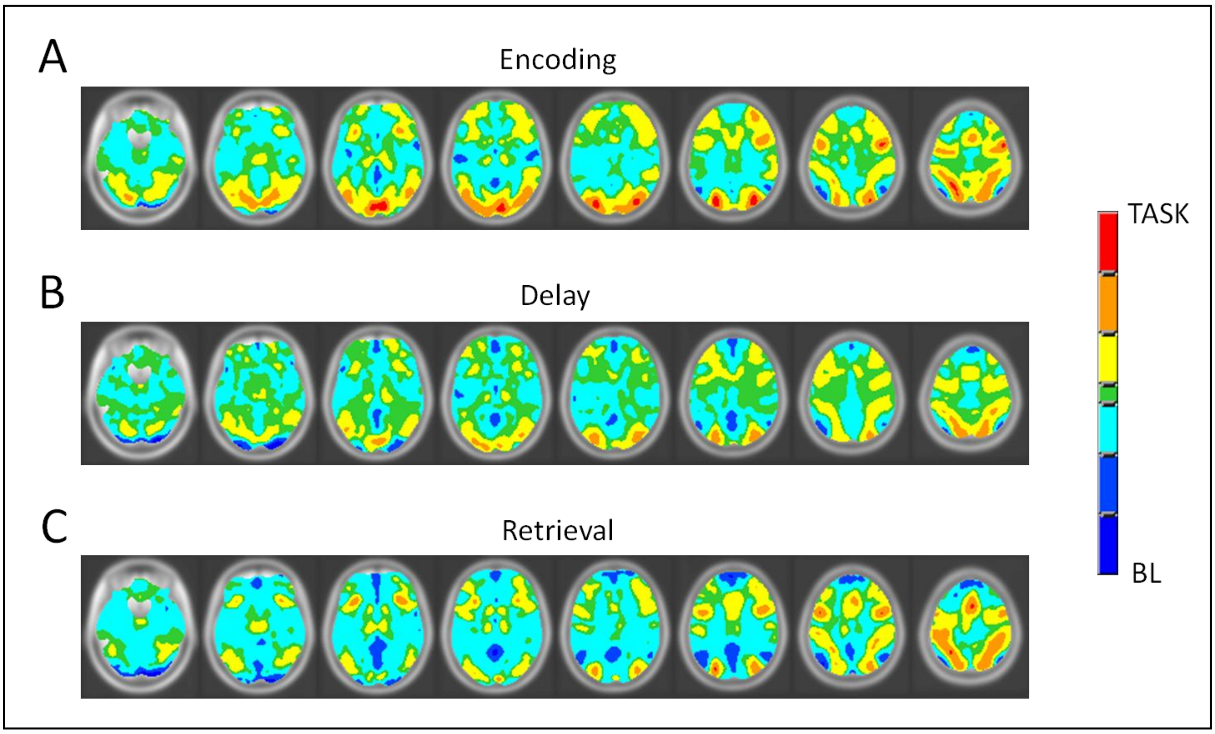


Figure S 1: Whole-brain GPC distribution maps for classifiers discriminating between task and baseline for each WM component (non-rewarded trials). A: Encoding B: Delay, C: Retrieval. Maps were rescaled such that the absolute maximum coefficient score was +/-1. The magnitude of GPC coefficients provides a measure of the relative difference in BOLD activity between experimental classes (in the context of the entire pattern) and the sign favours the class with greater mean activity. A distributed fronto-parietal network can be observed for each WM component along with task-related deactivations (TRDs) in regions consistent with the default mode network (DMN)

Distribution maps: task versus baseline (rewarded)

Whole-brain distribution maps derived from the classifiers trained to separate each WM component from baseline using rewarded trials are presented in Figure S 2.

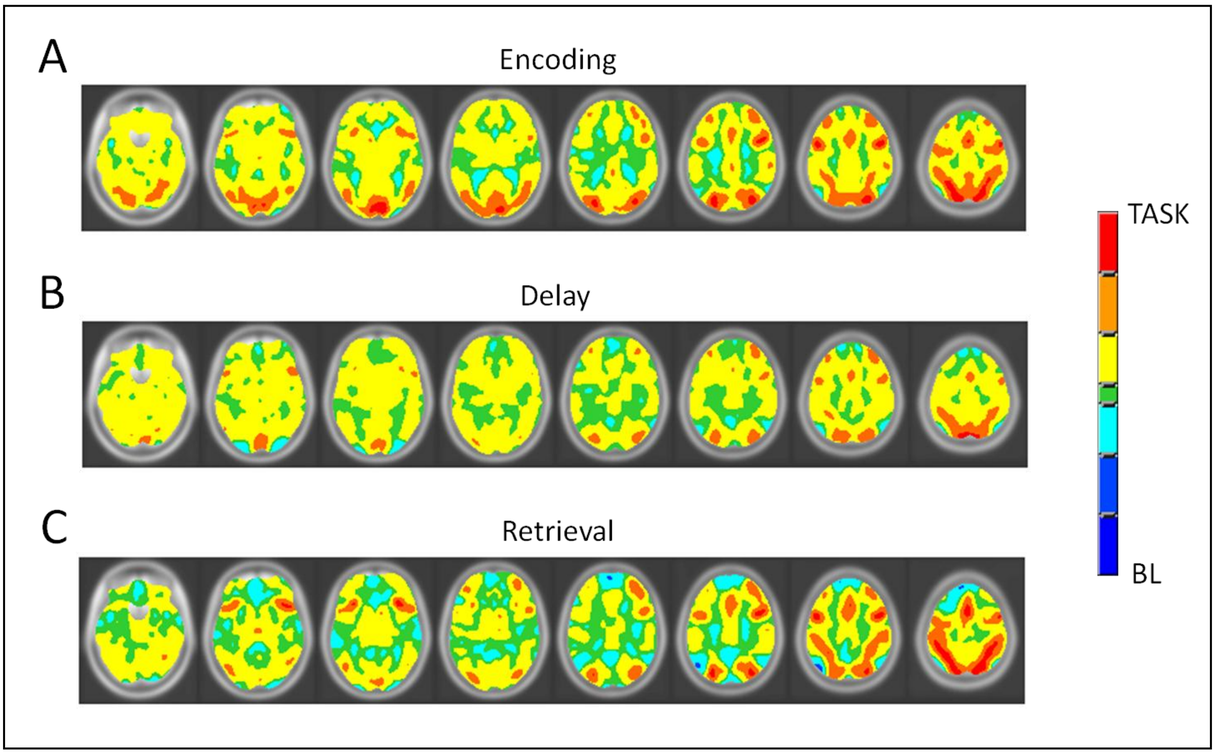


Figure S 2: Whole-brain GPC maps for classifiers discriminating between task and baseline for each working memory component (rewarded trials). Maps were rescaled such that the absolute maximum coefficient score was +/-1. A: encoding B: delay C: retrieval. Positive coefficients: task, negative: baseline. A distributed fronto-parietal network can be observed for each WM component

The distributed network of brain regions engaged by this task corresponds well to the activation foci described in previous studies where GLM analysis was employed (e.g. Curtis et al., 2004; Gibbs and D'Esposito, 2005), although the present analysis method is not limited to detection of focal effects.

Classification accuracy: whole-brain classifiers

For comparison with the GPC-RFE classifiers reported in the main text, we also report classification accuracy for comparable whole-brain GPC classifiers firstly for the classifiers discriminating between reward and control for each WM component and drug state (Figure S 3A) and secondly for the classifiers discriminating between drug conditions (Figure S 3B and C). The results are broadly similar, although differences from GPC-RFE classifiers are notable for MPH vs. PLC encoding on rewarded trials (66.67% GPC-RFE vs. 46.67% whole-brain GPC), ATX vs. PLC delay on rewarded trials (63.33% vs. 53.33%) and MPH vs. ATX encoding on non-rewarded trials (56.67% vs. 63.33%). The spatial map of the differential pattern for a sparse representation such as that derived using GPC-RFE allows inferences based on brain regions (or networks) which are derived in a principled manner. This contrasts with the whole brain mapping approach which gives no formal indication of the importance of specific brain regions to classification accuracy. For this reason, we have focussed on the GPC-RFE spatial representations in the main manuscript.

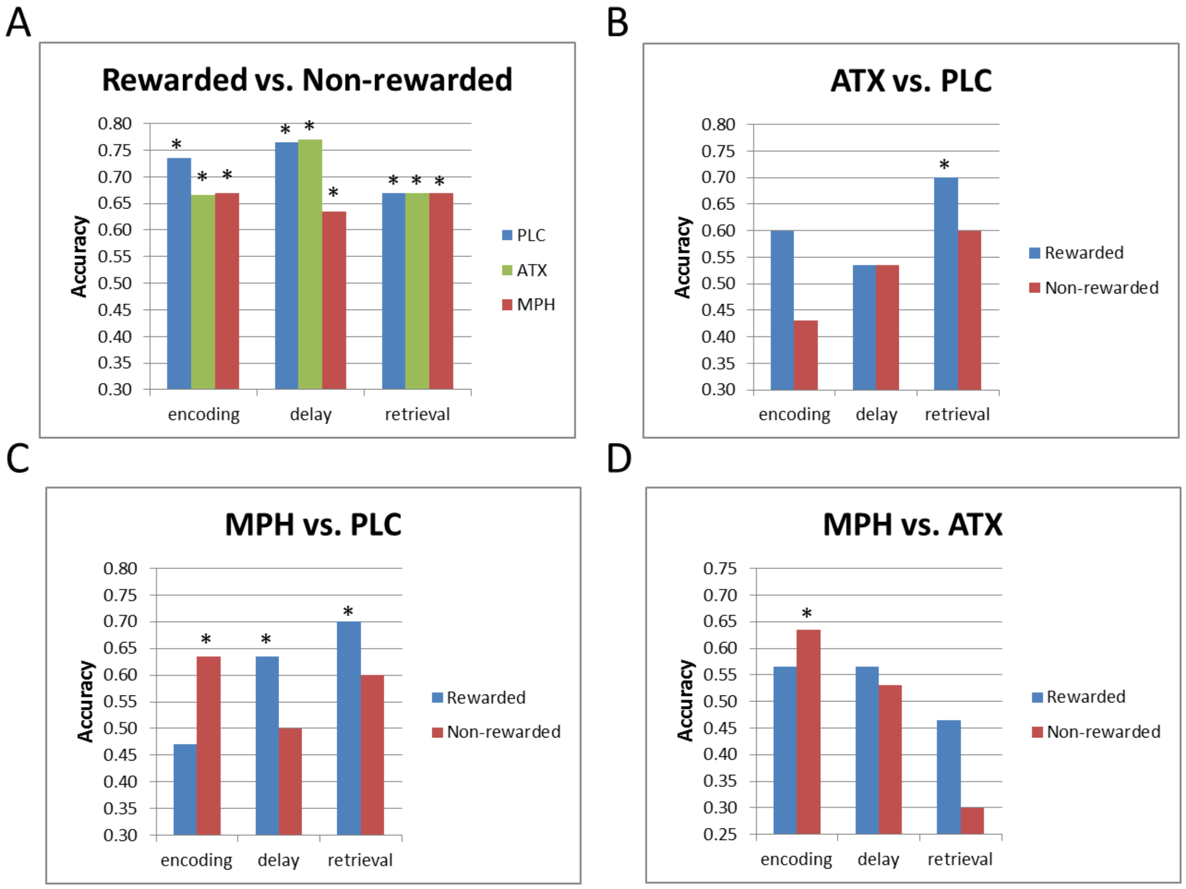
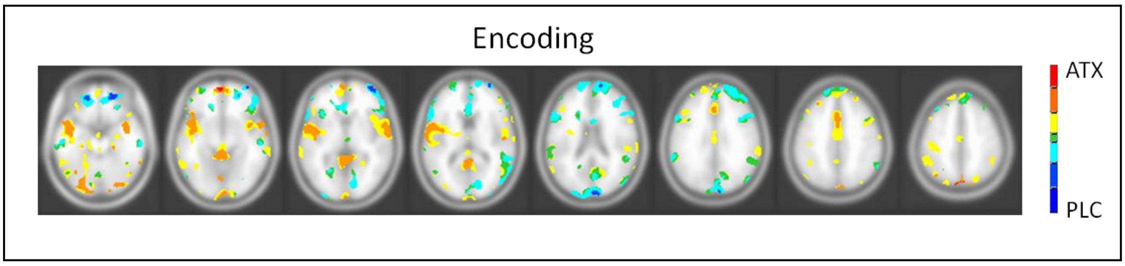


Figure S 3: Classification accuracy for whole-brain GPC classifiers for A: rewarded vs. non-rewarded trials, B: ATX vs. PLC, C: MPH vs. PLC and D: MPH vs. ATX. Asterisks indicate results significantly different from chance, i.e. 50% (p < 0.05, binomial test).

Distribution maps: ATX versus PLC (non-rewarded trials, encoding component)

The GPC-RFE distribution map for the classifier contrasting ATX and PLC for the encoding component of non-rewarded trials is presented in Figure S 4. We emphasize that this map was derived from a classifier that did not discriminate between classes above chance level (57% accuracy), so cannot be considered statistically sound (analogous to sub-threshold effects in a conventional univariate analysis). We present this map simply to illustrate that the differential pattern for ATX and PLC on the encoding pattern of non-rewarded trials shows some similarities with the map contrasting MPH and PLC on the encoding component of non-rewarded trials.

Figure S 4: GPC-RFE distribution maps for classifiers discriminating between ATX and PLC for the encoding WM component (non-rewarded trials). Note that this was derived from a classifier that did not exceed chance accuracy, so should be considered illustrative only. A distributed pattern favouring ATX can be observed that indicates that during encoding and in a non-rewarded context ATX weakly enhanced activity in some WM regions and weakly enhanced TRDs in DMN regions.



Supplementary References

Curtis CE, Rao VY, D'Esposito M (2004) Maintenance of spatial and motor codes during oculomotor delayed response tasks. Journal of Neuroscience 24:3944-3952.

Gibbs SEB, D'Esposito M (2005) A functional MRI study of the effects of bromocriptine, a dopamine receptor agonist, on component processes of working memory. Psychopharmacology 180:644-653.

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine Learning 46:389-422.

Herbert M, Johns MW, Dore C (1976) Factor-analysis of analog scales measuring subjective feelings before and after sleep. British Journal of Medical Psychology 49:373-379.

Kuss M, Rasmussen CE (2005) Assessing approximate inference for binary Gaussian process classification. Journal of Machine Learning Research 6:1679-1704.

Marquand A, De Simoni S, O'Daly O, Mourao-Miranda J, Mehta M (2010a) Quantifying the information content of brain voxels using target information, Gaussian processes and recursive feature elimination Qu. In: International Conference on Pattern Recognition. Istanbul, Turkey.

Marquand A, Howard M, Brammer M, Chu C, Coen S, Mourao-Miranda J (2010b) Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. NeuroImage 49:2178-2189.

Minka T (2001) A Family of Algorithms for Approximate Bayesian Inference (PhD Thesis). In. Massachusetts: MIT press.

Rasmussen C, Williams CKI (2006) Gaussian Processes for Machine Learning. Cambridge, Massachusetts: The MIT Press.